

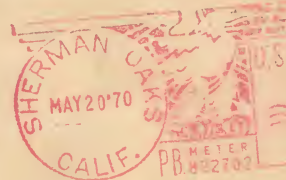


informatics inc.

5430 VAN NUYS BOULEVARD

SHERMAN OAKS / CALIFORNIA 91401

*Fulfilling The  
Computer's Promise*



A

ANOTHER **Datamation** READER INQUIRY

T NELSON SYS CONS

BOX 3

SCHOOLEYS MOUNTAIN

NJ 07870

NEWS

informatics inc.

5430 VAN NUYS BOULEVARD / SHERMAN OAKS / CALIFORNIA 91401 / (213) 783-7500

May 20, 1970

T. Nelson  
Systems Construction  
Box 3  
Schooleys Mountain  
New Jersey 07870

Dear Mr. Nelson:

Thank you for your request for information regarding the MARK IV File Management System, a general purpose product line designed to save up to 90 percent of your programming costs for business data processing on IBM System/360 and RCA Spectra 70 Computers.

MARK IV is marketed and supported as a proprietary software product of Informatics and is presently installed and operating in over 300 customer locations throughout the world. The MARK IV System provides facilities for file definition, file creation, file updating, retrieval, computation, reporting and subfile generation. Installation, education, field technical support, and continuing maintenance are provided in support of the product.

We are enclosing information about the MARK IV System. If you are interested in placing an order or desire additional information, we will be happy to discuss the system characteristics or business arrangements pertaining to MARK IV in more detail.

Very truly yours,

INFORMATICS INC.



S. R. Felderman  
National Sales Manager



FILE MANAGEMENT SYSTEM® The general purpose software product for business data processing

MARK

iV

FILE MANAGEMENT SYSTEM®

**CAN YOU AFFORD  
NOT TO HAVE IT?**

informatics inc.®



# WHAT HAVING MARK IV MEANS TO YOU!

At this moment, in over 200 installations around the world, the MARK IV File Management System is operating on IBM/360s. We can say with confidence that MARK IV is a proven product. It's been saving time and reducing costs for users in your area and in your industry for quite some time. If you'd like to know who some of the users in your area are, just write and ask us!

■ ■ ■ ■ ■ ■ ■ ■ ■ ■ *“In 1967, one programmer took six months to complete the COBOL programs required for the General Ledger Accounting System. The same programmer rewrote the application using MARK IV in eight days.”*

MARK IV is easy to use. It takes the tedium out of programming. Most of the job can be implemented in MARK IV by you in terms of your *application* requirements rather than in terms of the computer's requirements. 25% of what you used to program is handled completely automatically in MARK IV. Things like all of your input and output functions. You tell MARK IV what you want; MARK IV tells the computer what to do. Once you've defined your job, "programming" it in MARK IV is trivial compared to any other method.

■ ■ ■ ■ ■ ■ ■ ■ ■ ■ *"40% of our MARK IV jobs run the first time, 100% the second time."*

And it doesn't take forever to learn to use MARK IV either. In a few hours, most of MARK IV's capabilities can be taught, even to a user who is not familiar with computers or data processing. Everybody from a clerk to a president can learn productive use of MARK IV. In a few days of instruction, the experienced data processing individual is able to utilize all the power and flexibility of the MARK IV system.

■ ■ ■ ■ ■ ■ ■ ■ ■ ■ *"MARK IV was installed in November 1968. Within five months, 40% of all our jobs were running on MARK IV."*

MARK IV will do almost all of your business data processing tasks. Like Payroll, Accounts Receivable, Accounts Payable, Inventory, Sales Analysis, etc. You name it. And for quick response on those special one-time requirements, you've never seen anything like it.

[illegible]

Copyright 1969 Informatics Inc.

# MARK iV FILE MANAGEMENT SYSTEM®

# HOW MUCH DID NOT HAVING MARK IV COST YOU LAST YEAR?

■ ■ ■ ■ ■ ■ ■ ■ ■ ■ *"We planned to convert eight applications in three years with four programmers using COBOL. After 2½ months experience with MARK IV we converted all eight applications in six months with the same four programmers."*

■ ■ ■ ■ ■ ■ ■ ■ ■ ■ *"Phase I of the Material Accounting system took one year to design and eight months to program in COBOL. Phase II, which is three times larger, was completed in two months using MARK IV."*

These two quotations from satisfied MARK IV customers mark the two extremes of savings, one at a 6:1 advantage and the other at 30:1. We agree that most customers are closer to the 6:1 end of the range, but with that kind of saving nobody minds being at the bottom.

■ ■ ■ ■ ■ ■ ■ ■ ■ ■ *"It took less than 1000 MARK IV cards to tell the computer to do the same job (14 programs) that had required more than 10,000 COBOL cards."*

To make that 6:1 ratio a little more meaningful, let's assume that a programmer's time costs about \$200 per week. On the basis of \$10,000 a year for one programmer, we can say that one programmer can implement \$60,000 worth of MARK IV jobs in one year! That's a saving of \$50,000. Or, to put it another way, you can get six times as much programming accomplished. The increased output from just one programmer could save you enough to cover the entire cost of MARK IV in less than eight months. That means that if you had bought MARK IV eight months ago, it would have paid for itself by now and would already be producing additional profits for you.

Multiply what we've just said by the number of programmers you have working for you. That should give you an idea of how much it's costing you not to have MARK IV.

■ ■ ■ ■ ■ ■ ■ ■ ■ ■ *"We can count on a 10 to 1 reduction in programming time when using MARK IV instead of COBOL."*

Incidentally, our responsibility to you doesn't end when you buy MARK IV. Installation and support of MARK IV by Informatics includes:

- Reference Manuals, Operations Guides, User's Guides, Special Features Manuals, and Pracniques Handbooks;
- Updating service for all manuals;
- Training in the use of the system;
- Follow-up program of monthly technical visits;
- Periodic technical newsletters;
- Membership in the IV League, a users' organization to provide for further interchange of ideas, techniques, problems, and suggestions.

With this full customer service and support, the MARK IV File Management System is a powerful, dollar-saving asset to your data processing installation. To learn more about MARK IV, write to or call one of our Sales Offices listed on the back of this brochure. That may turn out to be one of the best moves you've ever made.

## SALES OFFICES

### LOS ANGELES

Informatics Inc.  
5430 Van Nuys Boulevard  
Sherman Oaks, California 91401  
(213) 783-7500

### BOSTON

Informatics Inc.  
675 Main Street  
Waltham, Massachusetts 02154  
(617) 894-0466

### CHICAGO

Informatics Inc.  
120 S. Riverside Place  
Chicago, Illinois 60606  
(312) 236-6466

### NEW YORK

Informatics Inc.  
65 Route 4  
River Edge, New Jersey 07661  
(212) 565-1163 or (201) 488-2100

### WASHINGTON

Informatics Inc.  
4720 Montgomery Lane  
Bethesda, Maryland 20014  
(301) 654-9190

### EUROPE

Informatics S.A.  
12a Route de Meyrin  
Geneva, Switzerland 1202  
Telephone 022/34.42.42

### AUSTRALIA

Independent Software  
of Australia Pty Ltd. (Affiliate)  
690 Springvale Road  
Mulgrave, Victoria 3170  
Telephone 560-6313

informatics inc.®



5430 VAN NUYS BOULEVARD / SHERMAN OAKS / CALIFORNIA 91401 / (213) 783-7500



FILE MANAGEMENT SYSTEM<sup>®</sup>

# TECHNICAL SYSTEM DESCRIPTION

JULY 1969

informatics inc.<sup>®</sup>



DOCUMENT NO. SP-69-810-9

PRODUCT DEVELOPMENT AND MARKETING DIVISION

5430 Van Nuys Boulevard / Sherman Oaks / California 91401

# SYSTEM DESCRIPTION

## TABLE OF CONTENTS

### PART 1. SYSTEM DESCRIPTION

<u>Section</u>		<u>Page</u>
1	INTRODUCTION TO THE SYSTEM . . . . .	1
2	DATA STRUCTURE . . . . .	3
3	FUNCTIONS . . . . .	6
4	STORAGE STRUCTURE . . . . .	23
5	OPERATIONAL ENVIRONMENT . . . . .	25

### PART 2. FEATURE LIST FOR THE SURVEY OF GENERALIZED DATA BASE MANAGEMENT SYSTEMS

*(Definition and format for the System Description as  
established by the CODASYL System Committee)*

<u>Section</u>		<u>Page</u>
1	INTRODUCTION TO THE SYSTEM . . . . .	25
2	DATA STRUCTURE . . . . .	26
3	FUNCTIONS . . . . .	29
4	STORAGE STRUCTURE . . . . .	35
5	OPERATIONAL ENVIRONMENT . . . . .	35
6	FEATURE AREA INTERACTION . . . . .	36

### *Acknowledgement*

*This document is adapted from "A Survey of Generalized  
Data Base Management Systems" prepared by CODASYL  
Systems Committee for the Conference On Data Systems  
Languages, Washington, D.C., May 1969.†*

†Copies are available for \$7.50 from:  
Association for Computing Machinery  
1133 Avenue of the Americas  
New York, N. Y. 10036

## PART 1. SYSTEM DESCRIPTION

### 1. INTRODUCTION TO THE SYSTEM

#### 1.1 IDENTIFICATION

The MARK IV File Management System is a general purpose file management system offered by Informatics Inc. of Sherman Oaks, California.

#### 1.2 STATUS

Release 8 of MARK IV was delivered in December 1968 and is currently in operation at more than 200 installations on a world wide basis. Release 9, scheduled for fourth quarter 1969, will feature improved performance by optimization of MARK IV code. Changes of this nature will have a minimal effect on previous work done by the user.

#### 1.3 SYSTEM BACKGROUND

As System 360 became a heavily used commercial computer, Informatics anticipated a market for a general purpose 360 file management system. Using experience gained from the earlier GIRLS system, and MARK I, II and III; the first external version of the functional specifications was published in July 1967.

The original design of the system was begun in late 1965. Release 3 of MARK IV (the first one obtainable) was available in February 1968; there have been five releases since then. Release 8 contains all of the functional capabilities as originally specified plus a large number of additional capabilities.

Optional special features providing additional functional capability may be purchased. Presently, Table Lookup and Indexed Coordinated File special features are available.

#### 1.4 MAJOR CHARACTERISTICS

##### 1.4.1 Data Structure Class

MARK IV provides a hierarchical data structure of up to nine levels. A single level data structure is a permissible subset of this structure.

##### 1.4.2 Generalized Processes Provided

The system provides facilities for file definition, file creation, file updating, interrogation, reporting and subfile generation.

##### 1.4.3 Language Type

The language is non-procedural.

##### 1.4.4 Language Form

Tabular forms are provided to perform the processes listed in 1.4.2.

##### 1.4.5 Storage Structure Class

The user can specify either a fixed length or variable length sequential storage structure or a fixed length indexed sequential storage structure.

##### 1.4.6 Modes of Use

MARK IV operates basically as a batch processing program; it can be used for remote job entry (RJE) processing.

##### 1.4.7 Hardware Environment

With the exception of the 360/20 and 360/44, the system can operate on any 360 configuration with one disk and any acceptable combination of peripherals. The Disk Operating System version of MARK IV requires 32K of core; the Operating System/360 version requires 128K of core.

##### 1.4.8 File Media

Files may be stored on tape or any 360 direct access device.

##### 1.4.9 Operating System Requirements

The system can run under the following two 360 operating systems:

- Disk Operating System (DOS)
- Operating System/360 (OS)

## 1.5 OVERALL PHILOSOPHY

The chief goal of MARK IV is to facilitate batch commercial data processing.

The use of tabular forms is intended to make basic information retrieval and maintenance functions easier for the commercial systems user who previously had to use a programming language. System default conditions on many forms are options intended to permit users to specify a minimum amount of information to accomplish data processing tasks. As many of these tasks can be batched together as required; the system informs the user of any errors, and at all times attempts to continue processing.

MARK IV is independent of OS and DOS release changes.

## 1.6 DOCUMENTATION

The following documentation is available from Informatics Inc.

- Reference Manual
- Special Features Manual
- Operations Guide
- User's Guide
- Pracniques Handbook

Internal system documentation is not publicly available.

## 1.7 SYSTEM DESCRIPTION

The primary function of the MARK IV File Management System is to provide the ability to manipulate files of data. The description of these files is independent of the files themselves. The structures and format of a file and the records (entries) within that file are defined to the system and stored in a dictionary. The transactions which

are used to create or update data files are likewise defined to the system and stored in a dictionary. These definitions identify the data that will update the file and specify the updating action to be performed.

After the files and their transactions have been defined, file maintenance can be performed. When the user specifies that a particular file maintenance is to be invoked, the system reads the master file, reads the transactions, and does the updating.

Once files have been created, information requests can be made. These requests are used to select entries from a file, select specified data from the entries for computation and logical processing and specify the desired output. This output takes the form of reports, intermediate result files, subsets of the original file, or combinations of all of these. In addition, the system has the ability to process multiple input files simultaneously; in a single run, the system can read 5 input files, generate 13 output files, and up to 255 different reports.

Requests that are to be repetitively used can be batched and stored in a system catalog as a request group; such a grouping is referred to as a cataloged job and may be subsequently invoked by specifying the request group name. In addition, each cataloged job can be modified by batching additional requests with it. For example, ad hoc requirements can be combined with a cataloged job for processing, thus alleviating the need for multiple file passes. If any requests contain errors, they are not processed, and further do not impact the processing of other valid requests.

MARK IV operates in a batch mode, serving two classes of users. The experienced user can apply the full capability of the system to accomplish his data processing tasks. The inexperienced user can use the request capability portion of the system to make ad hoc inquiries — he need only know the names of the data items and the master file they belong to.

## 2. DATA STRUCTURE

### 2.1 ITEMS

#### 2.1.1 System's Term For Items

The MARK IV term for item is *field*.

#### 2.1.2 Item Naming

A unique alphanumeric name, one to eight characters long, is assigned to each item within a file. Optional column heading text can be defined and formatted for each item named for use in subsequent reporting. If desired, these can be suppressed or modified when reporting.

#### 2.1.3 Item Data Types

Since MARK IV is a processing program run under DOS or OS/360, all 360 data types are valid. These are:

- EBCDIC Character String
- Zoned Decimal
- Packed Decimal
- Fixed Point Binary
- Floating Point Binary

A Table Lookup facility is provided for the decoding of data items. Decoded results may be used for processing and/or reporting.

#### 2.1.4 Data Variability

Depending on item data type, the range of item *fixed lengths* are:

- |                         |               |
|-------------------------|---------------|
| ● Character Strings     | 1 – 255 bytes |
| ● Zoned Decimal         | 1 – 15 bytes  |
| ● Packed Decimal        | 1 – 8 bytes   |
| ● Fixed Point Binary    | 1 – 4 bytes   |
| ● Floating Point Binary | 4 bytes only  |

#### 2.1.5 Multiple Valued Items

A repeating group (see 2.2.1 and 2.2.3) consisting of one item has one unique value for

each occurrence of the group. The system treats such a multiple valued item as a repeating group.

#### 2.1.6 Sub-Items

An item can be divided into sub-items by defining it more than once within a single file definition. For example, a 4-byte item DATE can be divided into two 2-byte sub-items YEAR and MONTH by defining the same 4-byte item in two different ways. In addition, different data types can be specified for the multiple definitions.

### 2.2 GROUPS

#### 2.2.1 System's Term For Group

The MARK IV term for group is *segment*. A segment is a collection of related items.

#### 2.2.2 Group Structure

Ninety-nine different types of groups (of items) within an entry may be defined using a group number from 1-99.

#### 2.2.3 Group Relationships

Level numbers specify the hierarchical relationship between groups. An entry can have up to nine levels of hierarchy. The first group is assigned level number 1 and subordinate groups are assigned level numbers 2 through 9, according to their relationship to the first group. Only one group is permitted at level 1, but there may be up to 98 groups at any lower level.

Groups at levels 2 through 9 may be repeating groups. For fixed length entries, these repeating groups occur a fixed number of times for each entry. For variable length entries, repeating groups may occur a variable number of times for each entry. To control the occurrences of variably repeated groups, the system maintains a count item, defined by the user, which automatically counts the occurrences of subordinate groups.

Figure 1 presents an example of a three level variable hierarchical entry with five groups. In Figure 2, this entry is mapped into a hierarchical graph.

EMPL. NO.	NAME		JOB TITLE		CLASS	DIVISION	DEPT
GROUP	HIRE DATE	DEPEND.	SOC. SEC. #	COUNT FIELDS			
				School	Jobs		
				2	1		

Level 1 Segment Type 1

SCHOOL #1	ADDRESS	YEARS	HONORS	DEGREE COUNT	REMARKS
				1	

Level 2 Segment Type 2

DEGREE	GPA	DATE

Level 3 Segment Type 3

SCHOOL #2	ADDRESS	YEARS	HONORS	DEGREE COUNT	REMARKS
				0	

Level 2 Segment Type 2

PREVIOUS EMPLOYER # 1		ADDRESS	TITLE	REF. COUNT	SALARY
				2	
TERM. DATE	START DATE				

Level 2 Segment Type 4

REFERENCE # 1	TITLE	YEARS KNOWN	EVALUATION

REFERENCE # 2	TITLE	YEARS KNOWN	EVALUATION

Level 3 Segment Type 5

Figure 1. Example of a Three-Level Variable Hierarchical Entry (Shaded Fields Indicate Group Keys)

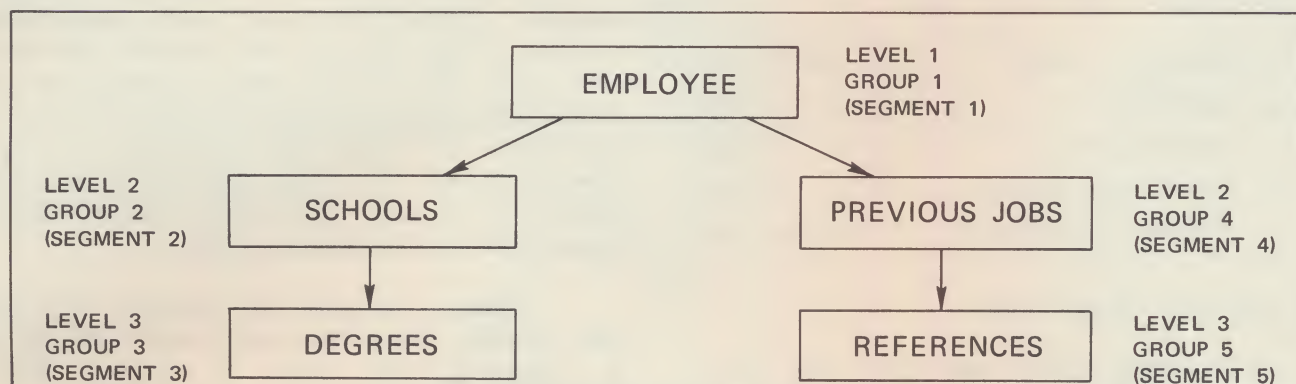


Figure 2. Tree Structure Diagram of the Three-Level Hierarchical Entry of Figure 1

#### 2.2.4 Group Identification

Groups are divided into one or more items containing values. Each group is assigned key items at file definition time indicating those items whose values uniquely define the group to its parent group. Up to three items in any one group can be designated for this purpose. They are numbered from 1 to 3 in order of major to minor significance. At least one item in a group must be a key item.

#### 2.2.5 Types Of Groups

There is only one group type within the system. This group type is a collection of related items as defined in 2.2.1 above. The system allows non-repeating groups; groups which repeat a fixed number of times; and groups that repeat a variable number of times.

### 2.3 ENTRIES

#### 2.3.1 System's Term For Entry

The system's term for entry is *record*.

#### 2.3.2 Entry Types

Files may be composed of entries of a single physical type. By defining a file more than once (renaming and redefining items within the file) many logical entries can correspond to one physical entry.

#### 2.3.3 Entry Identification

Entries consist of one or more hierarchically related groups, each identified by a key. The group key at the apex of the structure identifies each entry. Files which are updated must have unique entry keys; interrogations, however, may be specified on files containing duplicate entry keys.

### 2.4 FILES

#### 2.4.1 System's Term For File

The system's term for file is *file*.

#### 2.4.2 File Types

MARK IV does all its processing against *master files*. In creating and maintaining master files, users input *transaction files* containing transaction entries previously defined to the system. Those transactions not meeting predefined criteria may be placed in a *transaction reject* file. Entries deleted from a master file during a file maintenance run may be written in an *audit file* available to the user. During interrogation, up to three files, related by keys, can be used as auxiliary files while processing against a master file. These are referred to as *coordinated files*. Selected entries or specified items from selected entries from an interrogation can be placed in *subfiles* for further processing by MARK IV or another outside system. The same entries or items may be placed in *report files* if they are to be sorted and/or reported.

#### 2.4.3 File Identification

Each file must have one unique alphanumeric name, one to eight characters long. There is no MARK IV limit to the number of files that can be accommodated within the system.

### 2.5 DATA STRUCTURE GENERALIZATION

None.

### 2.6 DATA SECURITY

No automatic security is provided by MARK IV. In defining a file more than once, however, by specifying the availability of different items under each name, security at the item level is possible. This security is administered by the user.

### 3. FUNCTIONS

#### 3.1 LANGUAGE FORM

MARK IV is a completely tabular language. Twelve different forms are provided to accomplish all processing. Table 1 lists these forms and their functions.

#### 3.2 DATA STRUCTURE DEFINITION

The File Definition form is used to describe a data structure to MARK IV. Input from the form is processed by the system and placed in a dictionary on direct access storage. Several file definitions may be stored for one file, each one for a different requirement.

##### 3.2.1 Definition Of Data Items

All items in a file are assigned unique item names of one to eight alphanumeric characters.

Automatic table lookup result items are also defined on the File Definition form (see 3.8).

##### ■ Definition of Data Item Types

A data item type is specified for each item on the File Definition form. If none is specified, an EBCDIC character string is assumed.

##### ■ Definition Of Item Length Limitations

Fixed item lengths are specified for each item on the File Definition form. See 2.1.4 for ranges of permissible lengths.

##### ■ Definition Of Multiple Valued Items

Multiple valued items are implicitly defined by being the sole member of a repeating group.

##### 3.2.2 Definition Of Groups

Groups are defined by the definitions of the items they are composed of.

The group an item belongs to, its position within that group, and the group's hierarchical level are specified when the item is defined.

If the item is intended to be a group key, or a count item for a subordinate group appearing a variable number of times, or if it is a member of a group that repeats a fixed number of times, it is so designated.

##### 3.2.3 Definition Of Entries And Entry Types

Since files are restricted to contain entries of the same type, an entry definition is synonymous with the file definition.

##### 3.2.4 Definition Of Files And File Types

A file is named and its structure declared by filling out the File Definition form. This information is stored in a system dictionary which contains both file and transaction dictionaries. A file type is thus distinguished by the dictionary its name appears in. The transaction reject file, audit file, subfile, and report file are all output files and thus are not defined by the user.

When the user constructs a file definition, he has the capability of selecting any one of three glossary listings to list the file's contents and structure. An example of a file definition and glossary are given in Figure 3.

##### 3.2.5 Definition Of More General Data Structures

None.

##### 3.2.6 Definition Of Security

See 2.6.

##### 3.2.7 Data Validation

Minimum and maximum values of alphanumeric or numeric data for file creation are specified on the Transaction Definition form (file creation is accomplished by updating a null file with a transaction file). In the example given in Figure 13, EMPL NO may have a value between 1 and 4000.

Table 1. MARK IV Forms and Functions

FORM	FUNCTION
Information Request (IR form)	Entry selection; report specification; simple format and title specification.
Use Cataloged Request (CR form)	Invocation of stored processing requests.
File Definition (FD form)	Data structure definition; storage structure definition.
Transaction Definition (TD form)	Definition of transactions used to update master files.
Processing and Record Selection (ER form)	Entry selection; specification of arithmetic and logical operations on data items.
Output Content Specification (Rn form)	Specification of data for contents of reports and subfiles generated from a request.
Output Format Specification (En form)	Specification of the format of reports and production of subfiles.
Title (Tn form)	Specification of report titles, prefaces, comments, and Free Form report formats.
Temporary Field Definition (TF form)	Definition of temporary items used to perform certain processing functions.
Catalog Maintenance (CT form)	Storage or deletion of processing requests.
Run Control (RC form)	Specification of master files, transaction files and report files for a run, and run-dependent parameters.
Table Definition (TB form)	Definition of tables used by the Table Lookup Special Feature.

Some validation of data item types is done while processing a file — MARK IV outputs diagnostic messages if the S/360 operating system is unable to operate on an item because of an incorrect type specification.

### 3.2.8 Revision Of Data Definition

An entire file definition may be deleted by specifying DELETE and the file name on the File Definition form.

Portions of an existing file definition may be deleted or changed by specifying an item delete on the File Definition form followed by a redefinition of the item if desired. Possible changes are the:

- Data type and length of an item
- Location of an item within a group
- Group an item belongs to
- Identification of a particular group (the group key)

The contents of the file itself are not changed. Transactions make the data conform to the new definition.

### 3.3 STORAGE STRUCTURE DEFINITION

The storage structure is defined on the File Definition form by the entry format indicated. If index sequential is specified, entries of fixed length are organized as an index sequential data set on a direct access device. Sequential file storage structure is assumed for all other entry formats. Entry block size can be specified on the form. In the OS version of the system, this specification may be overridden by the OS Job Control Language.

In the example given in Figure 3 a sequential file has been specified with variable length entries.

**MARK IV**  
FILE MANAGEMENT SYSTEM™

**FILE DEFINITION**

FILE NAME: PERS FILE IDENT. FD DECK I.D. 73 OF 80

FILE IDENT. 11 DELETE? 19 GLOSSARY A RECORD FORMAT V RECORD SIZE 992 RECORDS PER BLOCK 1000 BUFFER SIZE 1000

FILE NAME	FORM CODE	FIELD NAME	FIELD NO.	FIELD LOCATION	FIELD LENGTH	FIELD TYPE	FIELD TYPE FOR EDIT	DECIMAL PLACES	THIS SEGMENT	OUTPUT EDIT	COLUMN HEADING TEXT
PERS		LOS/S.NMBR	11	160	9	1					
		LADDRESS	11	79	23						ADDRESS
		LNAME	11	1	45						
		LOLASTNAME	11	1	15						

DETAILED GLOSSARY BY NAME FOR FILE DEFINITION - PERS

FILE IDENTIFICATION = PERS  
NUMBER OF SEGMENTS IN FILE = 5  
NUMBER OF FIELDS IN FILE = 60

RECORD FORMAT = VARIABLE BLOCKED  
RECORD SIZE = 992  
BLOCK SIZE = 1000

\*\*\*\*\*  
\* SEGMENT 1, LEVEL 1 \*  
\*\*\*\*\*

SEGMENT OCCURS N TIMES = 1  
SEGMENT SIZE = 282  
NUMBER OF FIELDS IN SEGMENT = 44

KEY FIELD 1 = S/S.NMBR TYPE = C LENGTH = 9

FIELD NAME	FIELD TYPE	FIELD LOCATION	FIELD LENGTH	DECIMAL PLACES	COUNT FIELD FOR SEGMENT	EDIT CODES ( ) ( ) ( )	OUTPUT EDIT LENGTH	LINE NUMBER	COLUMN HEADING
ADDR	C	79	58				58		
ADDRESS	C	79	23				23	1	*** ADDRESS ***
B-DA	C	173	2				2		
B-DATE	C	169	6				6		
B-MO	C	171	2				2		
B-PLACE	C	175	30				30	1	*** BIRTH PLACE ***
B-YR	C	169	2				2	2	
BUSPHONE	P	277	6				15		
CITY	C	102	15				15	1	*** CITY ***
CITYSTAT	C	102	35				35		

Figure 3. File Definition and Detailed Glossary (Type A)

### 3.4 INTERROGATION

MARK IV interrogation facilities are provided on two different user levels. For requests requiring minimum user specification, the Information Request form can be used to:

- Select entries that meet specified criteria.
- Print a report containing data from selected entries, with specified totals and subtotals.
- Sort retrieved data in either ascending or descending order before printing.
- Define page titles.
- Specify the height and width of paper and vertical spacing.
- Produce a report consisting of summary information only.
- Control spacing between columns of data.
- Count the number of data values in any or all columns.
- Print maximum, minimum, and/or average values for any or all columns at various levels.

Four additional forms extend the above capabilities, enabling the user to compute with data items and constants, read multiple related input files, produce multiple subfiles, and vary the formats and titles of reports. The extended request also allows for percent and ratios, provides an override edit capability, and allows the use of partial fields for selection reporting. The forms and their specific functions are:

- Processing and Record Selection – selects entries and entry substructures according to specified criteria, and allows specification of arithmetic and logical operations on data items in the files.
- Output Content Specification – specifies the data for contents of reports and subfiles to be generated from a request.
- Output Format Specification – specifies the format of reports and production of subfiles.
- Title – specifies report titles, prefaces, comments, and Free Form report formats.

Up to 255 interrogation requests can be batched together for one processing run.

#### 3.4.1 Selection Criteria

##### ■ Atomic Conditions\*

##### ■ ■ Comparative Conditions On Item Value

Comparative conditions are expressed on both interrogation forms as:

Operand A: Relational Operator: Operand B

If the Information Request form is used:

Operand A is the name of the item in the master file to be accessed and processed, and Operand B may be the name of another item in the same file, or it may be a constant.

With use of the Processing and Record Selection form, Operand A may be specified as any of the following:

- Item name in the new master file
- Item name in the old master file
- Item in the first, second, or third coordinated file
- A temporary item (see 3.8)
- A flag (see 3.8)

Operand B may be any of the above plus a character string or decimal constant. When Operand A and Operand B are being compared, differences in length, type, and scale are processed automatically. For numeric items, decimal points are automatically aligned.

Six relational operators can be used with either form:

- Equal – EQ
- Not Equal – NE
- Less Than – LT
- Greater Than – GT
- Less Than or Equal – LE
- Greater Than or Equal – GE

---

\*See Part 2, Section 3.4.1.1

## ■ ■ Existence Conditions

The existence of data items of variable length entries may be tested for using the comparative test: data item = data item. If the item is non-existent the test will fail.

Since core storage is reserved and initialized for non-existent repeating groups of fixed length entries, the existence test data item = blank or data item = 0 can be made. Note, however, that the test is invalid if zero or blank is a legal value of the item being tested.

## ■ Item Conditions

AND/OR connectors are used to specify two or more atomic conditions on any one entry item. Logic levels are used to specify up to nine levels of logical nesting of atomic conditions.

Figure 4 shows how logic levels are used to logically connect the atomic conditions expressed on an item.

## ■ Entry Selection Criteria

AND/OR connectors are used to build up the total conditions for entry selection expressed on two or more items.

Figure 5 gives an example of how simple AND/OR logic for entry selection would be expressed on the Information Request form. AND/OR logic is handled identically on the Processing and Record Selection form.

## ■ Weighting Of Conditions

This capability is not automatically provided by the system. It can be accomplished, however, by storing weights in temporary items (see 3.8) and performing logical tests against these items.

### 3.4.2 Data Extraction

Data may be extracted from the old master file, the new master file, any coordinated master file (up to three) or any data items temporarily stored in core during any one processing run. Extracted data can be placed in subfiles for further processing, or in report files for reporting.

## ■ Generalized Extraction Features

Any item whose definition appears in the file definition dictionary may be extracted and also used as a sort key.

## ■ ■ Sorting

On either the Information Request form or the Output Content form, up to nine items can be designated as sort keys for input to an ascending or descending sort. The number 1 designates an item as the major sort key, the number 2 for the next most significant item, and so on. By concatenating these items after each sort, and using a new file definition, there is no limit to the number of items on which a set of extracted numeric or alphanumeric data can be sorted.

In the example of Figure 6, the sort items in order of minor to major significance are GROUP, DEPT, and DIVISION.

## ■ ■ Item Properties Which Are Extractable

By extracting the count item in a parent group, the user can determine the frequency of occurrence of a subordinate group (in a variable length entry structure). For example, if schools attended is a subordinate group to employee, then the schools attended count item of the employee group gives the number of occurrences of the schools attended group, and hence, the number of schools attended for that employee.

This is the only item property which is extractable.

## ■ ■ Discrete Extraction Sets Per Selection Criteria

Up to nine reports per selection criterion and/or ten subfiles per selection criterion can be produced on any combination of S/360 output devices supported by the Basic Index Sequential Access Method, the Queued Sequential Access Method, or the Queued Index Sequential Access Method.

Select records in which the field SALARY has the value of 1500 or (alternatively) select records in which SALARY has the value equal to or greater than 2000 and equal to or less than 4000. These criteria can be represented as follows:

STATEMENT: SALARY = 1500 or (SALARY  $\geq$  2000 and  $\leq$  4000)

The "or" statement not enclosed in parentheses is logic level zero; the "and" statement within the parentheses is logic level one. These selection criteria are entered in the record selection section as follows:

RECORD SELECTION									
SEQ. NO.		LOGIC LEVEL	CONNECTOR	FIELD NAME A	OPERATION	BLANK, C OR D	FIELD NAME B OR CONSTANT		
9, 10	11, 13	14	15	17	24, 25, 26	27, 28	35, 36, 48		
PR	1,0			SALARY	EQD	1500			
PR	2,0			SALARY	GED	20,00			
PR	3,0	1	A	SALARY	LED	40,00			
PR									

Figure 4. AND and OR Logic Level Example

RECORD SELECTION									
SEQ. NO.		LOGIC LEVEL	CONNECTOR	FIELD NAME A	OPERATION	BLANK, C OR D	FIELD NAME B OR CONSTANT		
9, 10	11, 13	14	15	17	24, 25, 26	27, 28	35, 36, 48		
PR	1,0			EXPER	EQD	5			
PR	2,0		A	MARISTATE	EQC	MARRIED			
PR									

Indicates an AND condition: Selects employees with 5 years experience AND who are married.

RECORD SELECTION									
SEQ. NO.		LOGIC LEVEL	CONNECTOR	FIELD NAME A	OPERATION	BLANK, C OR D	FIELD NAME B OR CONSTANT		
9, 10	11, 13	14	15	17	24, 25, 26	27, 28	35, 36, 48		
PR	1,0			EXPER	EQD	5			
PR	2,0		0	MARISTATE	EQC	MARRIED			
PR									

Indicates an OR condition: Selects employees with 5 years experience OR who are married.

Figure 5. Samples of AND and OR Logic

REPORT SPECIFICATION																	
			SEQ. NO.	NO. OF SPACES BEFORE COLUMN	FIELD NAME	SORT		BREAK		SUMMARIES							
9	10	11	13	14	15	17	24	27 SEQUENCE	28 DESC. ?	29 CONTROL	30 SUBTITLE	31 TOTAL	32 CUM.	33 COUNT	34 MAX.	35 MIN.	36 AVG.
R	1		10			DIVISION		1		1							
R	1		20			DEPT.		2		2							
R	1		30			GROUP		3		3							
R	1		40			SALARY						3					
R	1		50			DEPEND.						2					

In the report specification section above, control break numbers 1, 2 and 3 are assigned to fields DIVISION, DEPT., and GROUP, respectively. Summary action "TOTAL" is indicated for fields SALARY and DEPEND(ents), each field noted with the control break level at which summaries are to be taken. In the case of SALARY, the 3 in the "TOTAL" column indicates that summaries are to be taken for all value changes in fields numbered 3 or less. The 2 opposite DEPEND indicates that total summaries are to be taken for all value changes in fields numbered 2 or less. Thus, DEPEND totals are ignored for GROUP. The output from the above specifications appears below: Note that salary totals are given when GROUP, DEPT. and DIVISION values change and that DEPEND totals are given when DEPT. and DIVISION values change.

NOV 26, 1968			(TITLE)			PAGE 1		
			DIVISION	DEPART- MENT	GROUP	SALARY MTH	DEPEN- DENTS	
			A	101	10	\$2,180	4	
						1,050	2	
						925	1	
GROUP	TOTAL	A	101	10	\$4,155			
					15	\$1,800	3	
						1,950	5	
GROUP	TOTAL	A	101	15	\$3,750			
DEPT.	TOTAL	A	101		\$7,905		15	
				108	20	\$1,000	1	
						750	0	
						600	0	
GROUP	TOTAL	A	108	20	\$2,350			
DEPT.	TOTAL	A	108		\$2,350		1	
DIVISION	TOTAL	A			\$10,255		16	
			B	125	35	\$1,350	6	
						900	2	
GROUP	TOTAL	B	125	35	\$2,250			
DEPT.	TOTAL	B	125		\$2,250		8	
DIVISION	TOTAL	B			\$2,250		8	

Figure 6. Control Break, Summary, and Totals Example

## ■ Report Capability

If the Information Request form is used for interrogation, a printed report can be specified using the Report Specification portion of the Information Request form. If the interrogation facilities of the Processing and Record Selection form are used, the Output Content form is used to specify data for reports. Typically, each Output Content form requires a corresponding Output Format form (as described in 3.4.2) except:

- When an entire master file entry is selected on the Output Content form.
- When the user designates all default options (blanks) on the Output Format form, only the Output Content form need be used.

The Title form can also be used in conjunction with these forms.

Data selected for reporting are temporarily stored in report files. These files may be sorted before the report phase.

## ■ ■ Content Lines

If the Information Request form is used, the Report Specification portion of it is used to specify how data extracted from entries which meet the selection criteria is to appear in the report. The following can be specified:

- Items to be printed
- Items whose values may be totaled and averaged
- Items for which the maximum and/or minimum values can be ascertained and printed and items for which the number of values can be counted.
- Positioning of columns for the report
- Sorting of data in ascending or descending order
- Items which control the taking of summaries (control break items)

In addition, a summary report can be requested with all detail lines suppressed.

Figure 6 gives an example of a specified report.

If the Output Content form is used to specify output report data, the reporting capability is expanded to include the following:

- Data items selected from one entry can be specified to print on more than one line by using the END LINE capability. An example of this is given in Figure 7.
- A ratio or percentage may be taken on specified totals at specified control breaks. Figure 8 shows the use of this capability.
- Items can be used as control breaks on sort items without actually printing on reports.
- Partial items can be used for printing.
- An edit other than the one specified at file definition time can be used.

## ■ ■ Titles

Using the Title form a printed preface may be specified to begin on the second unnumbered page of the report, after the page containing requestor ID information. Figure 9 shows the use of the preface facility.

## ■ ■ Heading Lines And Footing Lines

Using the Title section of the Information Request form, a heading line can be specified to print at the top of each report page, centered between the report date, if any, and the page number.

Using the Title form, heading lines normally print at the top of each page. They will, however, begin on the second print line if:

- Date or page number are specified on the Output Format form to appear in the middle of the page.
- The first line of the page title is so long that it interferes with date and page number.

Figure 10 gives examples of page titles produced by the title form.

# NORMAL PRINTOUT

PRINTOUT																																																					
REQUEST NAME								FORM CODE		SEQUENCE NO.		NO. OF SPACES BEFORE COLUMN QUALIFIER		FIELD NAME								END LINE?		NON-PRINT		SORT		BREAK		SUMMARIES																							
1								8		9		10		11		13		14		15		16		17				24		25		26		27		28		29		30		31		32		33		34		35		36	
MAILEMP2								R I		1		0				FULLNAME																																					
								R I		2		0				ST-ADDR																																					
								R I		3		0				CITY-ST																																					
								R I		4		0				ZIP																																					

12/26/68

## FULL NAME

## STREET ADDRESS

## CITY AND STATE

## ZIP CODE

G UTT	1628 VIA ZURITA	PALOS VERDES ESTATES, CALIF	90274
R STONE	8242 SHIRLEY AVENUE	RESEDA, CALIF	91335
R WHITE	9943 BERTRAND AV	RESEDA, CALIF	91335
F BRADDOCK	18237 SUGARMAN ST	TARZANA, CALIF	91356
W CUTLER	991 EMERSON DRIVE	THOUSAND OAKS, CALIF	91360
H JACOBSON	3025 CHARLINDA ST	WEST COVINA, CALIF	91790

# END LINE PRINTOUT

E PRINTOUT																							
REQUEST NAME		FORM CODE	SEQUENCE NO.	NO. OF SPACES BEFORE COLUMN QUALIFIER	FIELD NAME	END LINE?	NON-PRINT	SORT		BREAK	SUMMARIES												
								SEQUENCE	DESCENDING?		CONTROL	SUBTITLE	TOTAL	CUMULATIVE COUNT	MAXIMUM	MINIMUM	AVERAGE						
1	8	9	10	11	13	14	15	16	17	24	25	26	27	28	29	30	31	32	33	34	35	36	
MAILEMP2		R I	10		FULLNAME Y																		
↓		R I	20		ST-ADDR Y																		
↓		R I	30		CITY-ST																		
↓		R I	40		ZIP								/										

12/26/68

## FULL NAME

## STREET ADDRESS

## CITY AND STATE

## ZIP CODE

G UTT	1628 VIA ZURITA	PALOS VERDES ESTATES, CALIF	90274
R STONE	8242 SHIRLEY AVENUE	RESEDA, CALIF	91335
R WHITE	9943 BERTRAND AV	RESEDA, CALIF	91335
F BRADDOCK	18237 SUGARMAN ST	TARZANA, CALIF	91356
W CUTLER	991 EMERSON DRIVE	THOUSAND OAKS, CALIF	91360
H JACOBSON	3025 CHARLINDA ST	WEST COVINA, CALIF	91790

Figure 7. End Line Example (Bottom) Contrasted to Normal Printout (Top).  
(Both examples are sorted by ZIP number.)

PAGE 1

Figure 8. Percent and Ratio ExampleFigure 9. Preface Specification Example

EXAMPLE 1: Short Title

NOV 26, 1968

PRODUCTION STATISTICS REPORT

PAGE 1

-----THE SECOND LINE OF TITLE MAY BE FULL WIDTH-----

EXAMPLE 2: Long Title

NOV 26, 1968

PAGE 1

THIS IS THE LONG FIRST LINE OF A PAGE TITLE

Figure 10. Page Title Example

By use of the Output Format form the title text specified with the Title form can be designated as footing lines to print at the bottom of each page.

■ ■ Other User Specified Text

The following may be specified:

● Subtitle Break

A change in the value of a control break item causes the new value of the item to be printed as a subtitle, after the printing of any specified summary information related to the last value of the item.

● Page Title Break

A change in the value of a control break item starts a new page after the printing of any summary information relating to the old item values. The new value of the item prints as a subtitle on the new page below the page title, but above the column headings.

The above capabilities are available using either the Information Request form or the Processing and Record Selection form. Figure 11 shows a subtitle break.

■ ■ Editing And Formatting

Formatting of report data fields can be specified for each item at file definition time. The editing consists of floating symbols [\$, +, -, or ( ), trailing symbols [+,-,), CR, or DB], and filling of lead zeros with any character. If any of these specifications are used, numeric items print with a comma separating digits in groups of three to the left of the decimal point. These commas can be suppressed if desired. Floating point numbers always print as ± .XXXXXXXXXE ± YY (no editing entry is specified for them). When no editing entry is specified for other numeric items, they print as follows:

- leading minus if negative
- leading blank if positive
- separating commas
- a decimal point followed by the number of decimal places specified

The file definition specified edit can be overridden at report time by using the Output Content form. Fifteen character places are provided to enter the edit as a "picture." This picture specifies the printing of additional characters or the truncation of the predefined ones, as shown in Figure 12.



Also at report time, the printing of any contiguous part of a character string field may be specified by using the Partial Field section of the Processing and Record Selection form.

#### ■ ■ *Derived Data*

Specified items can be designated as control break items on either the Information Request form or on the Output Content form. When values of these items change, the printing of specified summaries on other items is triggered. Nine levels of control break summaries can be specified in major to minor significance level. Summaries, as specified, print for each particular item whenever a control break occurs at a level equal to or greater than its own. The summary actions permitted for such items are:

- total since the previous control break
- cumulative total from beginning of report
- frequency count of printed items since the previous control break
- the maximum value printed for the item since the previous control break
- the minimum value printed for the item since the previous control break
- the average of all printed values that have occurred for the item since the previous control break

Figure 6 shows a control break example with the "TOTAL" summary action specified.

Using the Output Content form a ratio or percentage may be computed at control breaks from specified totals.

#### ■ ■ *Other Report Facilities*

Additional report formatting capability is provided by the Free Form facility. Free Form specifications are entered on the Title form. They permit the user to print titles anywhere on the page, print contents of fields in title lines, specify vertical placement of detail lines and position summary data anywhere below the last line of detail. It is typically used for formatting output on preprinted forms on which particular data must appear in specific locations. Free Form is used in conjunction with the Output Format and Output Content forms.

#### ■ *Extraction*

Up to ten files for use outside the system can be requested for each selection criterion. These files are 360 sequential files and can be read by any 360 processing program, including MARK IV.

#### ■ *System Triggered Outputs*

None.

### 3.4.3 *Invocation Of Predefined Interrogation*

Any interrogation can be saved by use of the Catalog Maintenance form which controls request and request group cataloging functions. Entries on this form allow request or request groups to be saved, deleted, replaced, inserted, listed, or dumped. Catalog requests may then be invoked by specifying the request name on the Use Cataloged Request card (form). Implicit modification of the request at run time may be accomplished by making use of temporary data fields.

### 3.4.4 *Other Features Of The Interrogation Process*

Ad hoc interrogations can be added to a group of cataloged interrogations for any MARK IV run. Thus, if a file is to be passed for a group of predefined interrogations, additional interrogation can be batched with these, avoiding the necessity of multiple file passes.

## 3.5 *UPDATE*

MARK IV updates master files by means of transactions. Transaction files are applied to master file records during a processing run to create or maintain master files. When transaction files are processed, the system automatically:

- Retrieves the transaction definitions from its dictionary.
- Determines the transaction type and its format.
- Matches the transaction with the appropriate master file entry, by values of specified items (normally the entry keys).
- Determines the processing to be performed from the action code in the transaction definition.

- Applies the transaction to the master file entry, item by item.
- Writes the updated entry onto a new master file (if required), after applying all transactions pertaining to that entry and after processing all requests.

Updating of selected master file entries based on the selection criteria logic described in 3.4.1 can be accomplished using the Processing and Record Selection form.

### 3.5.1 Selection Criteria

When the transaction method of updating is used, master file entries are selected based on the matching of values of specified items.

If the Processing and Record Selection method of updating is used, the interrogation selection criteria may be used to select entries for updating.

### 3.5.2 Update Specification

When the Processing and Record Selection form is used, the update specification filled in takes the form:

Operand A: Arithmetic or Replacement  
Operator: Operand B → Operand C

The arithmetic operators may be addition, subtraction, multiplication or division. Operand A and Operand B are as described in 3.4.1.

The update specifications described below refer to the transaction method of updating initially described in 3.5.

#### ■ Entry Level

To create new master file entries, a correct file definition for the desired entry must be in the file definition dictionary and a transaction definition, specifying the creation of a new master file record, must be in the transaction dictionary. Physical creation of the entry is accomplished when data is read into the system from the transaction file. Figure 13 gives an example of a transaction definition that will create an entry.

When a level 1 group is deleted, all groups subordinate to it are deleted. Thus, to delete a master file entry with transaction logic, a delete level 1 group transaction must be in the transaction dictionary, and all items that comprise the entry key must appear in the transaction data.

Entry deletion may also be accomplished by setting a system flag during processing. This flag tells the system not to write the entry in the new master file (see DELETE flag in Table 2).

#### ■ Group Level

A new group can be inserted in a master file entry if an insert group transaction was previously defined. The items of the transaction entry key and all group keys of a higher level than the one to be inserted must match their corresponding master file key items for the group data to be inserted. When a group is inserted in a variable length entry, the entry is enlarged to accommodate the new data. If a group is inserted in an entry with a fixed number of groups, space must have been provided at file creation time in the form of an empty group of the proper type.

Figure 14 gives an example of a transaction definition used for inserting a new group in a master file entry.

#### ■ Item Level

The following update actions on items can be specified:

- Adding transaction items to or subtracting them from master file items
- Clearing master file items to blanks or zeros
- Replacing master file items with transaction items (if the transaction item is blank, replacement may be suppressed)

#### ■ Intra-Item Level

Update specification on an intra-item level is possible by defining items more than once as subsets within one file definition.

# TRANSACTION DEFINITION

informatics inc.

PAGE \_\_\_\_ OF \_\_\_\_

DECK I.D.

IDENTIFIERS

TRANSACTION IDENTIFIER 1

TRANSACTION IDENTIFIER 2

TRANSACTION GROUP NAME: EMPL

FORM CODE: TD

DELETE? ☐ 29

GLOSSARY ☐ 30

MASTER FILE NAME: EMPLOYEE

CREATE RECORD? ☐ 39

INSERT SEGMENT? ☐ 40

RECORD FORMAT: F

RECORD SIZE: 80

RECORDS PER BLOCK: 1

BUFFER SIZE: 53

ACTION CODES

RECORD

CREATE ☐ C

DELETE ☐ D

SEGMENT

INSERT ☐ I

DELETE ☐ E

FIELDS

ADD ☐ A

SUBTRACT ☐ S

REPLACE ☐ P

REPLACE IF NON-BLANK ☐ R

REPLACE WITH BLANK OR ZERO ☐ B

TRANSACTION GROUP NAME	FORM CODE	TRANSACTION IDENTIFIER 1	TRANSACTION IDENTIFIER 2	DELETE	FIELD LOCATION	FIELD LENGTH	FIELD TYPE	FIELD NAME	ACTION CODE	DECIMAL PLACES	MINIMUM VALUE	MAXIMUM VALUE
EMPL	TD	1	4		1	4	P	EMPL NO.	C	0001	4000	
		2	6		6	20	P	NAME				
		3	26		26	6	P	HIREDATE				
		4	32		32	1	P	CLASS				
		5	33		33	20	P	JOB TITLE				
		6	53		53	11	P	S/S NO.				
		7	64		64	2	P	DEPEND				
		8	66		66	15	P	DIVISION				

Figure 13. Definition of a Transaction which will Create a Record with the TD Form

# TRANSACTION DEFINITION

informatics inc.

PAGE \_\_\_\_ OF \_\_\_\_

DECK I.D.

IDENTIFIERS

TRANSACTION IDENTIFIER 1

TRANSACTION IDENTIFIER 2

TRANSACTION GROUP NAME: EMPL

FORM CODE: TD

DELETE? ☐ 29

GLOSSARY ☐ 30

MASTER FILE NAME: EMPLOYEE

CREATE RECORD? ☐ 39

INSERT SEGMENT? ☐ 40

RECORD FORMAT: F

RECORD SIZE: 80

RECORDS PER BLOCK: 1

BUFFER SIZE: 53

ACTION CODES

RECORD

CREATE ☐ C

DELETE ☐ D

SEGMENT

INSERT ☐ I

DELETE ☐ E

FIELDS

ADD ☐ A

SUBTRACT ☐ S

REPLACE ☐ P

REPLACE IF NON-BLANK ☐ R

REPLACE WITH BLANK OR ZERO ☐ B

TRANSACTION GROUP NAME	FORM CODE	TRANSACTION IDENTIFIER 1	TRANSACTION IDENTIFIER 2	DELETE	FIELD LOCATION	FIELD LENGTH	FIELD TYPE	FIELD NAME	ACTION CODE	DECIMAL PLACES	MINIMUM VALUE	MAXIMUM VALUE
EMPL	TD	1	4		1	4	P	EMPL NO.	C			
		2	6		6	11	I	SCHOL				
		3	17		17	20	I	SCH-ADDR				
		4	37		37	9	P	YEARS				
		5	46		46	12	P	HONORS				

Figure 14. Definition of a Transaction which will Insert a Segment with the TD Form

Table 2. MARK IV Flags and their Applications

Flag	Application
TRAN	Indicates, with different values, the maintenance status of a master file entry and/or the rejection of a transaction.
CORD	Used to check if there was a match or no match condition for any of the three coordinated files.
TODAY DATE	Directs the system to provide the operating system date (in two possible formats) to the user.
PAGE	Directs the placement of page numbers when using Free Form output.
DELETE	Directs the system to delete this master file entry from the new master file.
OWN	Provided for Own Code users as a means of communication between user routines at Own Code exits.

### 3.5.3 System Triggered Updates

Fields which count variably occurring groups are automatically incremented or decremented.

### 3.5.4 Update Validation

Minimum and maximum allowable alphanumeric or numeric values can be specified for each transaction data item at transaction definition time.

Normally, if a specified transaction does not match a master file entry, the transaction is rejected and written in a transaction reject file if requested by the user. If, however, the DEFAULT CREATE entry is specified, a new entry is created with a single group at each of the levels having specified keys. If the items in the entry key match but the group key does not, and the DEFAULT INSERT group option has been specified, then new groups are created with specified keys beginning at the level where the match did not occur.

### 3.5.5 Invocation Of Predefined Updates

Transaction group names identify transaction definitions stored in the transaction dictionary. This dictionary information tells the system how the transaction data is to be matched and applied to the appropriate master file entry. A transaction group consists of one or more transactions and is invoked if the particular update is requested.

In the examples of Figures 13 and 14, the predefined transactions belong to the EMPL transaction group.

### 3.5.6 Audit Trail Facilities

Entries deleted from a master file during a file maintenance run may be placed in an audit file which may be used to print an audit trail. Rejected transactions may be placed in a transaction reject file available to the user. Other update changes are indicated by flags (see 3.8) and must be requested to be written in an audit file.

### 3.5.7 Other Update Facilities

None.

### 3.6 FILE CREATION

File creation is accomplished by creating entries as described in 3.5.2.

#### 3.6.1 Format Of Entry Level Input Data

Any legal MARK IV entry format can be used for input data.

### 3.7 GLOBAL FUNCTIONS\*

#### 3.7.1 Arithmetic Computation

Addition, subtraction, multiplication, and division can be used during the interrogation and updating processes for the following functions:

- Computing data for output to a tape, disk, or card, master file and subfile, or report, based on one or more selection criterion.
- Computing a new data item from an entry's data item for use in that entry's selection or a subsequent entry's selection.
- Performing conversions and other simple arithmetic operations on items.
- Performing computations with data items from consecutive entries.
- Performing computations with data items from coordinated files.

#### 3.7.2 Own Code

Own code routines, written in any 360 programming language may be linked into the MARK IV framework via an Own Code Hook.

While Own Code is only one module to MARK IV, it may consist of many subroutines to process the various activated hooks. At each of these hooks, the appropriate routine is provided with a parameter list containing all the information necessary for its processing.

### 3.8 OTHER FUNCTIONAL CAPABILITIES

#### 3.8.1 Temporary Fields

Temporary fields are defined explicitly on the Temporary Field Definition form or implicitly on the Processing and Record Selection form. They are used for the following processing:

- Perform computations
- Output computed values
- Communicate between interrogations and between entries
- Contain constant values for use in interrogation
- Store intermediate results

Explicitly defined temporary fields permit:

- The specification of column headings for temporary fields
- The specification of initial values for temporary fields
- The specification of field attributes for temporary fields
- The specification of output editing for temporary fields
- The definition of all temporary fields in one request of a run

Temporary fields which are implicitly defined are automatically created when they first occur, with the attributes of a base field. The temporary field may be Operand A, Operand B, or Operand C of the Processing and Record Selection request; the base field is the left-most defined operand preceding the temporary field. If there are no defined operands, the temporary field takes system default attributes.

#### 3.8.2 Flags

Flags are internal indicators generated by the system which the user can interrogate (or set) to indicate the existence of certain conditions. Table 2 lists these flags and their applications.

---

\*See Part 2, Section 3.7

### 3.8.3 Table Lookup Special Feature

MARK IV provides optional basic and automatic Table Lookup capabilities. Coded data items in files may be decoded and the decoded result used for processing and/or report output.

A basic table lookup operation is requested on a Processing and Record Selection form with a special operator. For automatic table lookup, a lookup result item is defined on the File Definition form. Reference to the lookup result item on an Information Request, Processing and Record Selection, Free Form or Output Content Specification form will cause the table lookup to be performed automatically. These lookup result items may be modified by dictionary maintenance runs.

The table itself is defined on a Table Definition form and is stored in the MARK IV dictionary by a dictionary maintenance run. It may be changed by subsequent dictionary maintenance runs. The contents of a table consist of a paired series of arguments (coded values) and results (decoded values).

The following table types are provided:

- Displacement table — has implied arguments, equal to the positions of the result values in the table list. An example would be a table of month numbers and names (e.g., 1 = JANUARY).
- Sequential search table — contains an unordered table list.
- Binary search table — has a sequential table list ordered by increasing argument value.

## 4. STORAGE STRUCTURE

MARK IV converts the user's data structure into a S/360 sequential or indexed sequential storage structure. These structures (files) are stored and maintained as data sets under DOS or OS Data Management.

### 4.1 ITEM LEVEL STORAGE REPRESENTATION

Items are internally represented in any allowable S/360 information format (see 2.1.3).

### 4.2 ENTRY AND GROUP LEVEL STORAGE STRUCTURE

Groups are organized within an entry in order of ascending group numbers. Item placement within a group is designated by a starting byte number and item length.

The maximum entry size is limited by the 360 Operating System Data Management.

### 4.3 FILE LEVEL STORAGE STRUCTURE

Fixed or variable length entries may be sequentially stored on tape or direct access files. Fixed length entries may be stored on indexed sequential direct access files.

### 4.4 MULTIPLE FILE STORAGE STRUCTURE

Inter-file linkages will be provided with the MARK IV ISAM Coordinated File special feature presently under development.

## 5. OPERATIONAL ENVIRONMENT

### 5.1 HARDWARE PARAMETERS

#### 5.1.1 Minimum Basic System

MARK IV has been implemented on the IBM S/360. The DOS System requires a 360/25 (32K) with at least one direct access device. The OS system requires a 360/40 (128K) with at least one direct access device. Larger 360 configurations can run MARK IV with increased speed, greater overlap of system operations, and can handle larger file entries.

#### 5.1.2 Storage Media

The system requires storage on any 360 direct access device. Files may be stored on any 360 serial or direct access device. All system and user tables must be on a direct access device.

#### 5.1.3 Terminal Equipment

As MARK IV is a batch system, this section is not applicable.

#### 5.1.4 Hardware Transferability

Within the S/360 family, MARK IV can operate on the Model 25 and up.

### 5.2 OPERATING SYSTEM PARAMETERS

MARK IV runs as a processing program under DOS or OS, and hence relies heavily on operating system (particularly data management) facilities. A goal of the system has been to remain as independent of new operating system releases as possible.

#### 5.2.1 Basic Required Operating System

Two versions of the system have been written to run under the following two 360 operating systems:

- Disk Operating System
- Operating System/360

The difference in the two systems is transparent to the user.

#### 5.2.2 Significant Features

Since MARK IV is designed to run in all 360 DOS or OS environments, all features of the operating system are important. For example, if MARK IV is to run in an MVT environment (multi-programming with a variable number of tasks), the interrupt facility of the operating system is critical to its functioning.

Neither version of MARK IV is transferable.

### 5.3 RESTART AND RECOVERY CAPABILITY

None.

### 5.4 SYSTEM OPERATION REPORTS

Statistical reports are provided on file sizes and usage.

### 5.5 OTHER ENVIRONMENTAL PARAMETERS

MARK IV makes certain environmental assumptions such as page height and page width on printed output. These parameters may be changed at each installation as desired.

## PART 2. FEATURE LIST FOR THE SURVEY OF GENERALIZED DATA BASE MANAGEMENT SYSTEMS

### 1. INTRODUCTION TO THE SYSTEM

#### 1.1 IDENTIFICATION

The name of the system is given together with some information on its origin.

#### 1.2 STATUS

The development status of the system (effective May 1969) is given. Planned extensions to the system are identified here and/or in the appropriate sections of the text.

#### 1.3 SYSTEM BACKGROUND

There should be some discussion of how the system came into being, what (if any) family of similar systems it belongs to and of any systems on which it is based.

#### 1.4 MAJOR CHARACTERISTICS

This is a brief description of the system in terms of the major characteristics listed below. The values shown for each characteristic are for illustrative purposes only; each characteristic is described briefly in a manner best suited to the system at hand.

##### 1.4.1 Data Structure Class

Single level, multiple level, hierarchic, graph, etc.

##### 1.4.2 Generalized Processes Provided

File definition, file creation, file updating, interrogation.

##### 1.4.3 Language Type

Procedural, non-procedural.

##### 1.4.4 Language Form

Tabular, string.

##### 1.4.5 Storage-Structure Class

Physical hierarchy, inverted, etc; orientation to retrieval or updating; user control of storage structure.

##### 1.4.6 Modes Of Use

Batch, on-line, etc.

##### 1.4.7 Hardware Environment

CPU's, peripherals, terminals.

##### 1.4.8 File Media

Disk, tape, etc.

##### 1.4.9 Operating System Environment

Specify any requirements.

### 1.5 OVERALL PHILOSOPHY

If possible, a statement of overall philosophy should be given which includes the objectives and the design rationale for meeting these.

### 1.6 DOCUMENTATION

A bibliography is given of the formal documentation available on the system and of any technical papers which discuss the system.

### 1.7 SYSTEM DESCRIPTION

Up to 400 words free form description of the system. This describes the common modes of operation of the system, specifying whether the users are on-line and whether or not they are knowledgeable in the use of computers. The description differentiates between the classes of applications for which the system is well suited, and those for which it is less well suited.

## 2. DATA STRUCTURES

This section describes the data as it is seen by the user of the system, without regard for any transformation that the data may undergo prior to storage in a *storage structure*. In some cases, the storage structure is identical to the data structure, in which case the description should be given here rather than in the section on storage structure.

The following defines terms as used in the data structure description. When these terms are used to indicate specific values rather than parts of the structure description, in order to avoid confusion they may be preceded by the phrases "instance(s) of" or "occurrence(s) of." The terms defined are limited to describing tree structures.

**ITEM** the term applied to the structural element which cannot be structurally subdivided and which may be associated with occurrences of values.

**GROUP** an association of zero or more items and/or zero or more groups. This implies that groups may be nested.

**ENTRY** the group at the apex of the tree structure. All other groups and items are included within the entry. (An occurrence of this is sometimes called a record, a term we avoid in order to forestall the confusion between "physical records" and "logical records.")

**FILE** the set of occurrences of the entry. Some systems permit multiple entry types in a single file.

### 2.1 ITEMS

#### 2.1.1 System's Term for Items

#### 2.1.2 Item Naming

This describes the system's facility for designating items, e.g., by character position within a group, by user-specified name, etc. Also included are facilities for assigning synonyms and column headings for printed output.

#### 2.1.3 Item Data Types

Items may have values in one of a number of data types, such as alphanumeric, integer, date, etc. This section describes the separate types provided by the system. Also included are facilities provided for different internal and external representations for a given item, e.g., coded items.

#### 2.1.4 Data Variability

This is the ability of the system to handle characteristics of an item, such as length, which may vary from one instance to another.

#### 2.1.5 Multiple-Valued Items

If a system has the ability for an item to have multiple values [apart from a facility for repeating groups (see 2.2)] or for it to have no value (null or non-existent value), it is described here.

#### 2.1.6 Sub-Items

Some systems permit an item to be divided into sub-items, sub-items into sub-sub-items, and so on. In a hierarchical system which allows this, a sub-item is not considered to be at a separate level in the hierarchy.

## 2.2 GROUPS

The concept of group is introduced to deal with the following situations:

- a. An association of items is considered as constituting a single logical entity, often with a name or other identification of its own. Such an entity is called a non-repeating group. For example, "number," "street," "city," and "state" might make up the group "address."
- b. In many cases an association of item values may occur more than once, e.g., the association "education" may consist of items "college," "degree," and "year," and each person may have several "educations." Such an association is called a repeating group. As a special case, a repeating group may contain just a single item, in which case it affords an alternative way of handling multiple-valued items (see 2.1.5).

- c. The hierarchical relating of items in an entry may be accomplished by "group-nesting," i.e., by permitting groups to contain other groups as constituents. Thus, the group "background A" might consist of the items "birthdate" and "birthplace," and the group "education." Alternatively, group constituents may be limited to items, and relating of items achieved by defining superior-inferior relationships between groups. E.g., in a different system the group "background B," consisting of items "birthdate" and "birthplace," might be declared superior to the group "education." A reference to "background A" will generally mean a reference to the items in background A and in its included groups; whereas a reference to background B" will include only the items in "background B" and not those in its subordinate groups.

#### 2.2.1 System's Term for Group

#### 2.2.2 Group Structure

This section describes any limitation on the number of groups, and details the method used to distinguish different groups.

#### 2.2.3 Group Relationships

This section describes the ability of the user to define relationships among groups, e.g., hierarchical relationships. In hierarchies, significant parameters are the number of levels permitted and the number of different groups permitted at each level. The description may include a graph-like diagram, for example one in which the nodes represent groups and the directed arcs represent the relationships among groups, e.g., "is superior to."

#### 2.2.4 Group Identification

For retrieval and updating purposes, each occurrence of a group may be required to contain an item or set of items which is unique within some set of such occurrences. Any such restrictions are described here.

#### 2.2.5 Types of Groups

In some systems there are several fundamentally different types of groups which are treated differently. Any such characteristics are described here.

### 2.3 ENTRIES

#### 2.3.1 System's Term for Entry

#### 2.3.2 Entry Types

Files may be composed of entries of a single type or of entries of multiple types; transaction files are a good example. This section describes the system's capability for accommodating files with multiple entry types; the method used to distinguish entry types; and limitations on number of types that can be accommodated.

#### 2.3.3 Entry Identification

For updating and retrieval purposes, entries often must contain an item or set of items which have a different value in each entry. Any such requirements, constraints, or options are described here. Also, indicated is any limiting factor on the number of entries in a file.

### 2.4 FILES

#### 2.4.1 System's Term for Files

#### 2.4.2 File Types

Some systems provide only a single type of file for all processing, while others provide a number of types which differ with respect to logical organization, or use within the system. For example, a system may have "master" files which collectively constitute the data base, and "transaction" files which are used to update the master files. The different types should be identified here for ease of reference later on.

#### 2.4.3 File Identification

This section describes the convention for identifying individual files within a multifile system. Also, indicated is the limiting factor on the number of files that can be accommodated in the system.

## 2.5 DATA STRUCTURE GENERALIZATION

This section describes any features provided by the system for accommodating data structures more general than those implied by the foregoing paragraphs. For example, a system may provide the data type "pointer" so that the user may establish explicit relationships among entries of the same or different files. Other systems permit the user to define such relationships but keep the pointers hidden from the user. The conventional procedure of using a normal item value from one file to locate a second item in another file is not considered to be a data structure feature.

## 2.6 DATA SECURITY

Many systems permit the user to define levels of security. These may be on a file entry, group, or item level and may be classified in terms of interrogation security and update security.

## 2.7 OTHER DATA STRUCTURE FEATURES

Any other features of data structure, as opposed to the process of defining data structure, are covered in this section.

### 3. FUNCTIONS

Certain functions can be performed on the data structure described in the previous chapter. The nature of the functions may depend upon whether the storage structure is apparent to the user. If it is, the functions expressible in the language may differ from the case in which the storage structure is not apparent to the user. If it is not apparent to the user, the language is usually oriented towards performing certain limited functions very easily.

The important functions considered include data definition, in which the data structure is defined (and possibly redefined). This holds whether the storage structure is apparent to the user or not. Other functions are those of interrogating and updating the data. The approach to those last two will differ between non-procedural free-standing systems and systems which are embedded in a procedural language.

If capabilities are common to two or more functions, they are described at the first appropriate point with subsequent back references.

#### 3.1 LANGUAGE FORM

Irrespective of the functions performed by the language, it must have a form. Some systems have highly tabular languages, others are free form within an 80 character limit and others are completely free form but use linguistic elements such as specific characters or words to terminate a specification. Some languages use a mixture of all techniques.

#### 3.2 DATA STRUCTURE DEFINITION

This section describes how the user defines the data structure of the system. All features discussed in Sections 2.1 through 2.7 are covered. Any restrictions on the way in which these definitions are made are mentioned — especially the sequence in which the various levels are handled.

##### 3.2.1 Definition of Data Items

This section describes how the lowest level in the data structure, namely a data item, may be

defined. Since items must always be given an identification, the process of assigning such identification is always described here.

##### 3.2.1.1 Definition of Data Item Types

If the system requires items to be given an item type, this is covered in this section.

##### 3.2.1.2 Definition of Item Length Limitations

Frequently item length is defined at the same time as the item type, especially in the case where the storage structure is apparent to the user and hence on this level the same as the data structure. In some cases the system requires the definition of a maximum length within which the item length may vary. Alternatively the length may not need to be limited for certain item types (usually alphanumeric) and the restriction on the entry length by physical storage considerations limits the sum of item lengths.

##### 3.2.1.3 Definition of Multiple-Valued Items

This section describes how multiple value capability is defined and any conventions called for by the system to designate specific instances of a multiple valued item.

#### 3.2.2 Definition of Groups

This section describes how multiple valued items and other types of data structure between the entry level and the item level are defined.

##### 3.2.3 Definition of Entries and Entry Types

This section describes how entry types are defined and how entries are identified within the file. Capability to handle several entry types in the same file usually only occurs in systems which are embedded in a host procedural language, since the non-procedural systems tend to be restricted to files containing entries all of the same type.

##### 3.2.4 Definition of Files and File Types

This section describes how a file is defined, how any notion of file type is introduced and how individual files are identified in a multiple file system.

### 3.2.5 *Definition of More General Data Structures*

This section describes how the more general data structures covered in Section 2.5 are defined.

### 3.2.6 *Definition of Security*

If the security is on an item level, then security may be defined when the item is identified or when it is assigned an item type. Alternatively security levels may be the subject of a special portion of the file definition process. If the security is on a file level, particularly in a time-sharing environment, then its definition may be an operating system function.

### 3.2.7 *Data Validation*

If the data validation facilities are an explicit system function, then how data validation is defined is described in this section. Only validation of data in the file is covered and not the validation of language specifications in the language nor the validation of the data for update transactions (see 3.4.4).

### 3.2.8 *Revision of Data Definition*

In several systems the data definition as described above may be revised or redefined on the basis of experience gained during a period of use of the data. This revision capability may be quite modest involving nothing more than a change to the data validation criterion or it may involve defining new data items for each record or even changing the hierarchical structure within the entry. File level revision must inevitably affect the storage structure. The revision features possible, how they are defined, and their effect on the storage structure are all subjects for discussion in this section.

### 3.2.9 *Other Data Definition Specifications*

Any other data definition specifications such as estimates of the activity on the file are described here. Specifications which influence or control the physical storage of the file, rather than its logical structure are included in 3.2.

## 3.3 *STORAGE STRUCTURE DEFINITION*

Control of storage structures is a specification which many systems regard as part of the file definition described above. This is especially true for systems in which storage structure is apparent to the user. If it is apparent on all levels, then the user has complete control over the storage structure. However, certain systems give the user an important measure of control over the file level storage structure. For instance, the user may be able to define certain data items as the subject of secondary indexes. In a non-procedural system this would usually cause the formation of such indexes during file creation, and their subsequent automatic maintenance during update.

## 3.4 *INTERROGATION*

In non-procedural self-contained systems, the function of interrogation is one most frequently automated. Interrogation typically consists of two parts, one to express a condition on the data in the file and the other to define which data must be copied out of the file into printed reports or mechanized files. This section indicates how interrogation is handled. How it is programmed in a conventional programming language or how it is automated in the framework of a non-procedural language should be described here.

### 3.4.1 *Selection Criteria*

Selection criteria facilities typically permit a complex condition to be expressed on each instance of an entry. The condition on the entry consists of conditions on one or more data items (usually the values of the item). There may be two or more logically connected conditions on an item. If the smallest indivisible condition is referred to as an atomic condition, then this may take numerous forms depending on the system. The different types of atomic condition possible and the way in which they can be connected to form the complete selection criterion are the chief factors of the feasibility of the facilities.

#### 3.4.1.1 *Atomic Condition*

An atomic condition is one expressed on a property of one or more data items. It is usually a relational condition on a single value using a relational operator such as equals, does not equal, greater than or less than. However, some systems

permit conditions on other item properties such as on the existence of a value, or a scanning of the characters in the value to see if they contain a specified sub-string. Other possibilities include the comparison of one item value with a value for another item and the comparison of some computed quantity based on data in the file with a specified quantity. This section must describe all forms of atomic conditions irrespective of how they may be connected with each other (see 3.4.1.2) and any weighting of conditions possible (see 3.4.1.4).

#### *3.4.1.1.1 Comparative Conditions on Item Value*

Since comparative conditions on item value are the most common kind of atomic condition, this section describes in more detail how such conditions may be expressed, including the forms of the two arguments in the comparison and of the relational operators used.

#### *3.4.1.1.2 Existence Conditions*

If it is possible for a condition to be expressed on the existence of a value for an item, this implies that the system will handle non-existent data. A discussion of this capability and the form for the existence condition is the subject for this section.

#### *3.4.1.1.3 Other Atomic Conditions*

Any type of atomic conditions not covered in the preceding two sections are described in more detail in this section.

#### *3.4.1.2 Item Conditions*

Two or more atomic conditions on the same item may often be logically connected (by AND or OR) without repeating the identification of the item. Other systems use different techniques to build up the total condition on the item. In several systems the conditions on the item may be nested to a certain depth. All facilities for expressing composite conditions on a single item of data are described in this section.

#### *3.4.1.3 Entry Selection Criterion*

The total conditions expressed on two or more items may be logically connected, either disjunctively or conjunctively, and possibly

associated with conditions on quantities computed from two or more item values. How complex such total conditions may be is the subject of this section.

#### *3.4.1.4 Weighting of Conditions*

Some systems permit (or even require) the user to assign a weight to each atomic condition or to a group of conditions and also to assign a threshold for the total conditions on the record. If the condition on the record is evaluated and "its total weight" found to exceed the threshold, then the condition is regarded as true. In some cases the total weight is retained and used to order the output. This section describes all facilities of this type which are available.

#### *3.4.1.5 Other Features of Selection Criteria*

Any other features of the way in which selection criteria are stated are described in this section.

#### *3.4.2 Data Extraction*

Data in an entry satisfying the selection criteria may be extracted and placed into either report form or into machine readable form (namely mechanized files), possibly after intermediate processing. If extraction may involve data from multiple files, it is discussed here. To permit meaningful comparison, the sections under this section define only the facilities for publishing data from an entry which has satisfied the selection criteria.

#### *3.4.2.1 Generalized Extraction Features*

In some systems there exist facilities for defining a choice of physical medium on to which the extracted data is to be copied. In this case there are features to be described which are independent of the choice of medium, for example which quantities may be extracted from the data file, whether such quantities can be used as sort key for sorting the data prior to its final output.

#### *3.4.2.1.1 Sorting*

Sorting extracted data is a fairly common capability among non-procedural systems. How sort keys are identified, and how many are permitted, whether the user has a choice of collating sequence and other facets of the sorting facilities are topics to be discussed in this section.

#### *3.4.2.1.2 Item Properties which are Extractable*

In all systems it is possible to extract data values for inclusion in the output report or file. It may also be possible to extract other properties of items, such as length, frequency, etc. What can be accomplished is the topic for this section. Capabilities which extend across items or entries are described in 3.4.2.2.6.

#### *3.4.2.1.3 Discrete Extraction Sets Per Selection Criterion*

Most systems allow the user to specify one report or file to be produced from the data entries satisfying the selection criterion. Some systems allow two or more outputs to be specified for the same selection criterion without respecifying the criterion. Any capability of this nature, including that of mixing output media, is described in this section.

#### *3.4.2.2 Report Capability*

Different kinds of reports may be requested by the user. This section includes only reports containing data from the file or derived from data in the file. It excludes any system reports about the system's own operation. It includes a description of reports produced for human perusal on media other than paper, for instance video consoles or plotters.

##### *3.4.2.2.1 Content Lines*

The way in which the main body of the report specified is defined in this section. This includes specification of data to appear at control breaks.

##### *3.4.2.2.2 Titles*

The facility to define a title for inclusion at the front of a report is described.

##### *3.4.2.2.3 Heading Lines and Footing Lines*

The facility to define lines of text for inclusion at the top and bottom of each page of the report and the facility to include data in the body of such text are described.

#### *3.4.2.2.4 Other User Specified Text*

The facility to include strings of text within the body of the report is described.

#### *3.4.2.2.5 Editing and Formatting*

The editing of output data is handled in such procedural language as COBOL. In some non-procedural systems the output format for an item must be defined at file definition time and is then always used. In other systems, a user definition when the report is specified may override the built-in format. Other systems give the user little control over the output format of data items.

#### *3.4.2.2.6 Derived Data*

Several systems give the user the capability to include certain well identified quantities in their output report, where the quantities are computed from data in the file. Typical of such quantities are sums and means, maximums and minimums. Other systems provide computational capability so that the output report can contain data computed by user specified procedures. This section contains a description of what derived data can be output and how this must be specified.

#### *3.4.2.2.7 Other Report Facilities*

If the system has other inherent capabilities for printing or displaying data in a form to be read by the human eye, then these should be described in this section.

#### *3.4.2.3 Extraction of Files for Use Outside the System*

Many systems have the capability to extract mechanized files which are formatted in such a way that they can be used outside the system, for example, as input to programs written in such languages as FORTRAN and COBOL. The capabilities, including the user control over the storage medium, are described in this section.

#### *3.4.2.4 Extraction of Files for System's Use*

If the system has capability for generating files, under user control, for use within the system, then this is described in this section.

#### 3.4.2.5 *System Triggered Outputs*

The system may have the facility to produce reports or files, using the facilities described in 3.4.2.2, at certain times or when criterion conditions occur in the data base. This section describes the facilities for specifying the time, the conditions and the value of the outputs. Again it does not include reports about the system's own operation but reports containing data in the data base, which are described in Section 5.

#### 3.4.3 *Invocation of Predefined Interrogations*

Some systems have the capability to store frequently used interrogations, comprising selection criterion and extraction specification, with the file. This interrogation may then be invoked either in its stored form or with explicit or implicit modification of its parts. Any such facilities should be described here both in terms of how and when the interrogation is stored and how and when it may be invoked.

#### 3.4.4 *Other Features of the Interrogation Process*

Any features not covered in the preceding section from 3.4.1 to 3.4.3 are described here.

### 3.5 *UPDATE*

Update is the process of changing the value content of the file or data base in accordance with the receipt of input messages frequently referred to as transactions. Such input messages may be those which define specific selection criteria or they may be messages requiring analysis by the system to determine the selection criteria to be applied.

Update must not be confused with the process of revising or redefining data definitions, validation criteria, value sets, etc., which is covered in 3.2.8. The process of specifying updates is analogous to that of specifying interrogations in that it is also a two part process. A condition must be satisfied and an action carried out. The process of specifying the condition may involve either a direct selection of a single entry in terms of the entry identifier or it may require a selection criterion as complex as that used in interrogation. Several systems recognize this similarity and use the identical language for specifying the update selection criterion.

#### 3.5.1 *Selection Criteria*

This section may be a back reference to 3.4.1. Any differences are discussed in this section.

#### 3.5.2 *Update Specification*

Given that the condition is satisfied and that an update action is to be carried out on an entry, then the action may be on several levels. This section surveys the level with a more detailed description in the following sub-sections.

##### 3.5.2.1 *Entry Level*

Insertion of new entries and deletion of current entries are covered in this section.

##### 3.5.2.2 *Group Level*

Insertion and deletion of groups are covered in this section.

##### 3.5.2.3 *Item Level*

Update action on items is covered in this section.

##### 3.5.2.4 *Intra-Item Level*

Capability for update actions defined within an item, possibly down to the character level, is covered in this section.

#### 3.5.3 *System Triggered Updates*

Some systems allow one input message or one update to trigger one or more other updates. Also, it is possible for updates to be triggered by time in a similar way as interrogations (see 3.4.2.5). All such facilities are covered in this section.

#### 3.5.4 *Update Validation*

Some systems permit a definition of a validation for the update action, while others use the validation criteria described in 3.2.7. How and when updates are validated are described in this section.

### 3.5.5 Invocation of Predefined Updates

This facility is analogous to the interrogation facility covered in 3.4.3. Some systems are oriented towards all updates being predefined in the form of a template to which values are added when the update is used. All such facilities are described in this section.

### 3.5.6 Audit Trail Facilities

This section describes capability for listing changes made during updating. Such a list is produced to permit the user to check the modifications to the data.

### 3.5.7 Other Update Facilities

Any update facilities not covered in the preceding sections are described here.

## 3.6 FILE CREATION

The file creation process, namely that of producing the initial instance of the file, may be handled by updating a null file or it may be handled as an appendage to the file definition process as described in 3.2.

Some systems do not have a process of file creation explicitly conceived, but they operate only on files which are already in mechanized form and which can be defined with the type of facilities described in 3.2. How initial file creation is achieved is described in this section. If the system has explicit facilities for reading data entries to be input to the system, then the restrictions in the input form are described in the following sub-section.

### 3.6.1 Format of Entry Level Input Data

This describes how the data values to be stored for a new entry must be formatted on the input medium. Any differences between file

creation and new entries for file update are noted. This section does not include details of the format for lower levels of update such as the item level which should be covered in 3.5.2.3 and 3.5.2.4.

## 3.7 GLOBAL FUNCTIONS

Some operations may be global to the system or at least to the interrogation and update facilities. For example, it may be possible to specify arithmetic computation involving the basic addition, subtraction, multiplication and division, and also derived arithmetic operations, such as roots, exponentiation and trigonometric functions. It may also be possible to transfer to other languages.

### 3.7.1 Arithmetic Computation

In the non-procedural systems, the interface with arithmetic computation facilities may be limited by system restrictions, while in systems embedded in procedural languages, such facilities will be equivalent to those of the host language. Any facilities which have not been covered in earlier sections such as 3.4.1.1.1, 3.4.2.2.6, and 3.5.2.3 are described here.

### 3.7.2 Own Code

Any ability or necessity to enter a procedural language (or code generated by its processors) is covered in this section as is any interface with assembly language.

## 3.8 OTHER FUNCTIONAL CAPABILITIES

Any capabilities not covered in the preceding sections from 3. to 3.7.2 are described in this section.

## 4. STORAGE STRUCTURE

The storage structure capabilities vary extensively among different systems. Some generalized systems consist of nothing more than facilities to define and operate on sophisticated storage structures feasible only in a direct access device. In other systems the user of the system need have no knowledge of the storage structure, which usually implies that he has no control over it.

Storage structures can be described on at least two levels. The organization of the data within a stored entry, of entries within a file and files within a multiframe system are examples of levels. Typical storage structure techniques may include the use of primary and secondary indexes, pointers, chains, etc.

### 4.1 ITEM LEVEL STORAGE REPRESENTATION

This would include the internal representation of discrete values (including null where appropriate) and also the storage of multiple occurrence items.

### 4.2 ENTRY AND GROUP LEVEL STORAGE STRUCTURE

The organization of values and possibly links, intra entry indexes or separators within the

stored representation of the entry are described in this section.

### 4.3 FILE LEVEL STORAGE STRUCTURE

Within the stored representation of a file, entries may be stored sequentially as is necessary on a sequential medium, such as magnetic tape. On a direct access device more intricate file level storage structures are not only feasible but also necessary to take advantage of the capabilities of the medium. Various indexing and chaining techniques are possible and they are described in this section.

### 4.4 MULTIPLE FILE STORAGE STRUCTURE

The representation of file linkage and indexes to files within a multiframe system are described in this section.

### 4.5 OTHER STORAGE STRUCTURE FEATURES

Any other features of storage structures not covered in the preceding sections are covered in this section.

\* \* \* \* \*

## 5. OPERATIONAL ENVIRONMENT

No data management system can be described completely independently of the environment in which it exists. This environment consists mainly of hardware and software, and these two facets are discussed separately.

### 5.1 HARDWARE PARAMETERS

All systems are implemented or being implemented on at least one hardware configuration and may be implemented on many. The description of the hardware parameters covers what is minimum and what is recommended.

#### 5.1.1 Minimum Basic System

This section lists the manufacturer and identification of the central processor configuration with specific information about any optional features of importance like core memory size, emulators, clocks, special RPQ's, etc.

#### 5.1.2 Storage Media

Different systems will permit or require different types of serial (e.g., tapes) or direct access (e.g., disks) storage devices. Where possible minimum number required is given for each type.

### 5.1.3 Terminal Equipment

This section is primarily for those systems that permit on-line use. Each type of remote console (i.e., typewriter, cathode ray tube display) that can or must be used is listed. The way in which this equipment is connected to the central processor is described.

### 5.1.4 Hardware Transferability

Some systems are designed to work on only one particular hardware configuration; others have as a goal complete machine independence. This section emphasizes the machine independence goals of the system and how well these goals are attained through use of higher order programming languages, special designs, etc.

## 5.2 OPERATING SYSTEM PARAMETERS

The reliance of data management systems on executive, control or operating systems varies widely. Some rely heavily on manufacturer's supplied operating systems, others only on programming language compilers, while some include within the data management system those functions normally performed by the operating system. This general section attempts to pick the appropriate spot on the spectrum for the system being discussed.

### 5.2.1 Basic Required Operating System

This section specifies by generally accepted nomenclature or reference to published documentation the operating system or systems under which the described system operates.

### 5.2.2 Significant Features

Every operating system has a large number of features only some of which are extremely important to the described system, such as file cataloging and handling, ~~I/O~~ access, interrupt processing, scheduling, etc. For each feature of significance, the specific options are discussed together with reasons why the feature is important.

### 5.2.3 Transferability Between Operating Systems

This section discusses the feasibility of transferring the described system to other operating systems.

## 5.3 RESTART AND RECOVERY CAPABILITY

This section describes the capability in the system for recovering and restarting.

## 5.4 SYSTEM OPERATION REPORTS

This section describes any capability in the system for producing reports about its own operation.

## 5.5 OTHER ENVIRONMENTAL PARAMETERS

This section covers any environmental characteristics that do not fit under hardware or operating systems.

\* \* \* \* \*

## 6. FEATURE AREA INTERACTION

Feature areas of certain systems may interact in such a manner that it is germane to the understanding of the system itself. Such interaction may be described here with the subsection specifications left open.

## SALES OFFICES

### LOS ANGELES

Informatics Inc.  
5430 Van Nuys Boulevard  
Sherman Oaks, California 91401  
(213) 783-7500

### CHICAGO

Informatics Inc.  
120 S. Riverside Place  
Chicago, Illinois 60606  
(312) 236-6466

### BOSTON

Informatics Inc.  
675 Main Street  
Waltham, Massachusetts 02154  
(617) 894-0466

### WASHINGTON

Informatics Inc.  
4720 Montgomery Lane  
Bethesda, Maryland 20014  
(301) 654-9190

### NEW YORK

Informatics Inc.  
65 Route 4  
River Edge, New Jersey 07661  
(212) 564-1258 or (201) 488-2100

### EUROPE

Informatics S.A.  
12a Route de Meyrin  
Geneva, Switzerland 1202  
Telephone 022/34.42.42

### AUSTRALIA

Independent Software of  
Australia Pty. Ltd. (Affiliate)

690 Springvale Road  
Mulgrave, Victoria 3170  
Telephone 560-6313

14th Floor, Penfole House  
88 Pitt Street, Sydney, NSW 2000  
Telephone 28-6304